Laboratorul 12.

Probleme

1) Singleton

Realizati o aplicatie in care veti construi o clasa **ShopSingleton** folosind design pattern-ul **Singleton**. Clasa va avea urmatorii membrii:

- name (String)
- products (o lista de produse)

In acest caz veti avea o clasa **Product** care descrie un produs, caracterizat prin pret, care va fi float si un nume, care va fi un String.

In **ShopSingleton** va exista o metoda void numita ***showProducts***, care va afisa produsele din magazin. Veti avea o clasa Test, in care veti avea implementate urmatoarele metode:

- addProduct(Product) - veti adauga un produs in magazin,
- removeProduct(Product) - eliminarea unui produs din magazin,
- getCheapestProduct() - va returna produsul cel mai ieftin din magazin

Si un main in care veti testa aceste metode.

2) Factory

Modificati exercitiul anterior astfel: clasa **Product** va fi abstracta si va fi extinsa de urmatoarele clase, care vor diferi printr-o metoda **getPriceReduced** de tip float, care va returna pretul redus al unui produs, depinzand de categoria acestuia:

- Book (15\%)
- Food (20\%)
- Beverage (5\%)
- Computer (10\%).

Clasa prin care se vor crea obiectele de tip **Product** se va numi **ProductFactory** si va avea o metoda **factory(String type, String nameProduct, float productPrice)** care va returna un obiect de tip **Product**, construit cu **productName** si **productPrice**, in functie de tipul de produs dorit de utilizator, specificat prin parametrul **type**.

3) Observer

Implementati un mini-sistem de notificare a utilizatorilor abonati la un canal Youtube. Veti implementa o clasa-subiect numita **Channel**, care va reprezenta un canal si o clasa-observator numita **User**, care va reprezenta un utilizator. Clasa **Channel** va contine o lista de utilizatori abonati la canalul respectiv, un membru de tip String ce va reprezenta numele canalului si urmatoarele metode:

- void subscribe (User user) - se va adauga un utilizator in lista de abonati
- void unsubscribe (User user) - se va sterge un utilizator din lista de abonati
- void notify (String notification) - se va trimite o notificare (mesaj de tip String) tuturor utilizatorilor din lista de abonati (de exemplu ca s-a incarcat un nou videoclip pe canal)

Testati clasele implementate intr-un main.

4) Strategy

Implementati un mini-sistem de calcul al pensiilor lunare in functie de anii de vechime si pe baza unui salariu lunar dat. Interfata **Strategy** va contine o metoda **calcul(int aniVechime, float salariu)** care returneaza un float. Veti avea de implementat clasele:

- TwentyStrategy, care calculeaza pensia dupa formula aniVechime / 20 * salariu
- ThirtyStrategy, cu pensia calculata dupa formula aniVechime / 30 * salariu
- FortyStrategy, cu pensia calculata ca aniVechime / 40 * salariu.

Creati o clasa numita Pensionar, care contine trei membri (int aniVechime, float salariu si Strategy strategy) si o metoda de tip float getPensie(), care va returna pensia calculata folosind strategiile enuntate anterior in urmatorul fel:

- daca 20 ⇐ aniVechime < 30, atunci se va folosi TwentyStrategy
- daca 30 ⇐ aniVechime < 40, atunci se va folosi ThirtyStrategy
- daca 40 ⇐ aniVechime, atunci se va folosi FortyStrategy

Testati functionarea sistemului intr-un main.

5) Visitor

Pornind de la exemplul prezentat in breviar la Visitor, implementati comenzile ls si cat pentru fisiere (clasele **Ls** si **Cat**).

Clasele **Fisier** si **Director** mostenesc o clasa abstracta **Repository**, care este de tip **Visitable**, iar clasa **Director** contine o lista de **Repository**-uri, care reprezinta fisierele si folderele din folderul respectiv.

Clasa **Director** va contine o metoda de tip void **addChild**(Repository rep), care va adauga un Repository in lista de Repository-uri din folder (aici puteti sa va folositi de clasa **File**, mai precis de una dintre metodele list(), care returneaza un array de String-uri de nume de fisiere si foldere din folderul curent, sau listFiles(), care returneaza o lista de obiecte de tip File (fisiere si foldere).

In clasa **Cat**, la visit(Fisier), veti folosi citirea din fisiere pentru a afisa continutul unui fisier.

poo/laboratoare/12.txt · Last modified: 2021/01/11 17:57 by carmen.odubasteanu